Modélisation 3D à partir d'images en utilisant plusieurs cartes de profondeur

Pau Gargallo, Peter Sturm

Modélisation 3D à partir d'images en utilisant plusieurs cartes de profondeur

Bayesian 3D Modeling from Images using Multiple Depth Maps

Pau Gargallo Peter Sturm

GRAVIR, Équipe MOVI, INRIA Rhône-Alpes
655 Avenue de l'Europe, 34334 St-Ismier CEDEX FRANCE
prenom.nom@inrialpes.fr

Résumé

*Cet article porte sur la reconstruction de la géométrie et de l'albédo d'une scène Lambertienne à partir d'images calibrées obtenues de points de vue écartés (*wide-baseline*). Afin de prendre en compte la totalité des données, nous proposons de représenter la scène comme un ensemble de cartes de profondeur colorées, une pour chacune des images d'entrée. Le problème est formulé d'une façon probabiliste comme un problème de* maximum a posteriori*. Les effets des occultations et des reflets spéculaires sont gérés en introduisant des variables cachées de visibilité qui signalent si les points du modèle sont visibles ou non dans les images. Les principales contributions de ce travail sont : la définition d'un* a priori *pour ces variables de visibilité qui tient compte des occultations géométriques et la définition d'un* a priori *pour l'ensemble de cartes de profondeur qui lisse et fusionne les cartes tout en tolérant de fortes discontinuités. Plusieurs exemples montrant l'efficacité et les limitations de la méthode sont présentés.*

Mots Clef

Stéréo dense, reconstruction multi-vue.

Abstract

This paper addresses the problem of reconstructing the geometry and albedo of a Lambertian scene, given some fully calibrated images acquired with wide baselines. In order to completely model the input data, we propose to represent the scene as a set of colored depth maps, one per input image. We formulate the problem as a Bayesian MAP problem which leads to an energy minimization method. Hidden visibility variables are used to deal with occlusions, reflections and outliers. The main contributions of this work are: a prior for the visibility variables that treats the geometric occlusion; and a prior for the multiple depth maps model that smoothes and merges the depth maps while enabling discontinuities. Real world examples showing the efficiency and limitations of the approach are presented.

Keywords

Dense stereo, multiview reconstruction.

1 Introduction

Nous nous intéressons au problème de l'extraction de modèles 3D de haute résolution d'une scène à partir d'une petite collection d'images. La reconstruction 3D à partir d'images a été largement étudiée dans le domaine de la vision par ordinateur. Beaucoup d'algorithmes on été proposés. Les différences entre eux sont principalement liées au modèle utilisé pour représenter la scène, à l'*a priori* sur ce modèle et à la méthode de recherche du meilleur modèle. Le type de représentation est un facteur très important qui conditionne fortement les avantages et les faiblesses des différentes méthodes.

Les méthodes qui utilisent des modèles volumiques comme les voxels [9, 2, 11] ou les ensembles de niveau [4], sont basées sur une discrétisation de l'espace en cellules et leur but est de déterminer la frontière entre les cellules pleines et celles vides. Elles peuvent utiliser beaucoup d'images prises de points de vue arbitraires. Toutes les topologies peuvent être reconstruites et la visibilité est gérée d'une façon géométrique. Néanmoins, la discrétisation initiale limite la résolution finale des modèles. On ne peut obtenir des modèles plus fins qu'en augmentant la résolution de la grille de voxels. D'un autre côté, les méthodes utilisant des maillages [7, 10, 20] peuvent, en théorie, adapter leur résolution pour mieux reconstruire les détails de la scène. Par contre, elles souffrent des problèmes liés à l'auto-intersection du maillage et à la difficulté de gestion des changements topologiques pendant le traitement.

Les cartes de profondeur sont une représentation qui a été, principalement, étudiée pour les cas d'une paire d'images prises de points de vue proches l'un de l'autre [1, 13, 8, 16, 19]. Ceci rend impossible l'obtention de résultats précis, et, donc, ces méthodes utilisent de forts *a priori* qui souvent introduisent du biais fronto-parallèle. Les résultats de ces méthodes ne sont pas des surfaces lisses et précises, mais des surfaces planes par morceaux. Récemment, une méthode de reconstruction utilisant une carte de profondeur et plusieurs images avec de points de vue écartés a été développée avec des très bons résultats [15, 14]. L'écart entre les points de vue permet des résultats très précis et l'utilisation d'une carte de profondeur élimine les problèmes de

discrétisation et des changements topologiques des autres modèles.

Ces avantages liés à la représentation par des cartes de profondeur nous encouragent à les considérer. Cependant, il est fréquent qu'une seule carte ne suffise pas pour représenter la totalité de la scène. Seulement les parties visibles dans la vue de référence sont modélisées. De plus, avec une seule carte de profondeur, il est difficile d'utiliser toute l'information disponible dans les images, car la reprojection des points de la carte sur les images d'entrée ne couvre en général pas la totalité des pixels. Il faut une carte de profondeur pour chacune des images d'entrée [18] pour pouvoir assurer que tous les pixels d'entrée seront utilisés et modélisés. Au lieu de calculer chacune des cartes de profondeur indépendamment et de les mélanger dans une étape de post-traitement [12, 14], nous calculons toutes les cartes en même temps. Ceci permet de traiter la visibilité d'une façon efficace et assure la cohérence entre les cartes retrouvées.

Dans [14], l'estimation d'une carte de profondeur à partir d'une série d'images est formulée comme un problème de *maximum a posteriori* (MAP) en utilisant le cadre de travail proposé dans [5] pour le problème de synthèse de nouvelles vues tout en montrant la relation entre les deux problèmes. Ici on reprend la même formulation du problème et on l'adapte au cas de plusieurs cartes de profondeur.

Les contributions de cet article sont les suivantes. Premièrement, une réflexion sur la formule utilisée pour calculer la vraisemblance et une modification conséquente. Deuxièmement, la définition d'un *a priori* pour les variables de visibilité qui utilise la géométrie de la scène. Et finalement, nous présentons un *a priori* pour l'ensemble des cartes de profondeur qui fusionne et lisse les cartes tout en permettant des fortes discontinuités.

1.1 Définition du problème

Notre but est de trouver une représentation 3D d'une scène à partir d'une collection d'images complètement calibrées (les paramètres intrinsèques et extrinsèques sont connus). Le modèle utilisé pour représenter la scène est composé d'un ensemble de cartes de profondeur colorées. Pour chaque pixel de chaque images d'entrée, nous devons inférer la profondeur et la couleur du point 3D correspondant à ce pixel.

2 Modélisation et estimation

Nous considérons le problème comme une recherche du *maximum a posteriori*. Les images d'entrée $\mathcal{I}$ sont considérées comme une mesure bruitée du modèle θ. Le modèle recherché est défini comme celui qui maximise la probabilité *a posteriori* $p(\theta|\mathcal{I}) \propto p(\mathcal{I}|\theta)p(\theta)$.

Nous commençons par décrire les variables significatives du problème dans la section 2.1. Après, dans la section 2.2, nous décomposons la probabilité conjointe de ces variables, déterminant ainsi les dépendances statistiques entre elles. Les sections 2.3, 2.4 et 2.5 décrivent la forme de chaque terme de la décomposition. Finalement, dans la section 2.6 nous présentons la méthode qui à été utilisée pour estimer le MAP dans notre implémentation.

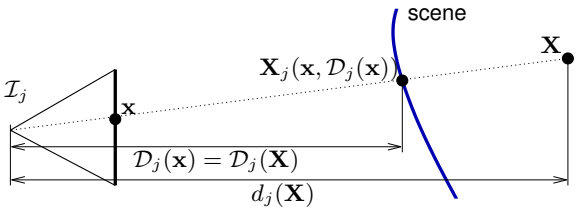

FIG. 1 – Pour un point 3D $\mathbf{X}$, $d_j(\mathbf{X})$ est sa profondeur relative à l'image $\mathcal{I}_j$. $\mathbf{D}_j(\mathbf{X})$ est la profondeur estimée du pixel qui est à la projection de $\mathbf{X}$ sur l'image $\mathcal{I}_j$, $\mathbf{x} = \mathrm{P}_j\bar{\mathbf{X}}$.

2.1 Les cartes de profondeur et de couleur et les variables de visibilité

L'ensemble des n images d'entrée est noté par $\{\mathcal{I}_i\}_{i=1..n}$. $\mathcal{I}_i(\mathbf{x})$ est la couleur du pixel $\mathbf{x}$ de l'image i, représentée dans un certain espace de couleurs (niveau de gris, RGB, etc.). Les caméras sont représentées par un ensemble de matrices de projection $\{\mathrm{P}_i\}_{i=1..n}$. Ces matrices sont de la forme usuelle $\mathrm{P}_i = \mathrm{K}_i(\mathrm{R}_i|\mathrm{t}_i)$ avec $(\mathrm{K}_i)_{3,3} = 1$. La profondeur d'un point $\mathbf{X} = (X, Y, Z)^\top$ par rapport à une caméra P_i est définie comme $d_i(\mathbf{X}) = (\mathrm{P}_i\bar{\mathbf{X}})_3$, où $\bar{\mathbf{X}} = (X, Y, Z, 1)^\top$. Réciproquement, si le pixel $\mathbf{x} = (x, y)^\top$ de l'image i a une profondeur d, alors les coordonnées euclidiennes du point 3D correspondant sont $\mathbf{X}_i(\mathbf{x}, d) = d(\mathrm{K}_i\mathrm{R}_i)^{-1}\bar{\mathbf{x}} - \mathrm{R}_i^\top\mathrm{t}_i$, où $\bar{\mathbf{x}} = (x, y, 1)^\top$.

Pour chaque pixel de chaque images d'entrée nous calculons sa profondeur et sa couleur. Les profondeurs sont stockées dans un ensemble de cartes de profondeur $\{\mathcal{D}_i\}_{i=1..n}$ et les couleurs dans un ensemble d'images idéales $\{\mathcal{I}_i^*\}_{i=1..n}$. $\mathcal{D}_i(\mathbf{x})$ et $\mathcal{I}_i^*(\mathbf{x})$ seront, donc, la profondeur et la couleur du point correspondant au pixel $\mathbf{x}$ de l'image i. Parfois, il sera plus intuitif de penser à l'ensemble des cartes de profondeur colorées comme une représentation du nuage de points 3D $\{\mathbf{X}_i(\mathbf{x}, \mathcal{D}_i(\mathbf{x})) : i = 1..n, \mathbf{x} \in \mathcal{I}_i\}$ et de traiter tous les points du nuage de la même manière, tout en oubliant leur origine (c'est-à-dire, la carte de profondeur par laquelle un point est paramétré).

Pour simplifier la notation, étant donné un point $\mathbf{X} = \mathbf{X}_i(\mathbf{x}, \mathcal{D}_i(\mathbf{x}))$ du nuage de points, sa couleur estimée $\mathcal{I}_i^*(\mathbf{x})$ est notée $C(\mathbf{X})$. La couleur de son projeté sur les autres images est notée $\mathcal{I}_j(\mathbf{X})$ au lieu de $\mathcal{I}_j(\mathrm{P}_j\bar{\mathbf{X}})$. De façon similaire, on écrira $\mathcal{D}_j(\mathbf{X})$ au lieu de $\mathcal{D}_j(\mathrm{P}_j\bar{\mathbf{X}})$. Il est important de remarquer que ceci est la profondeur estimée du *pixel* qui est à la projection du point 3D $\mathbf{X}$ sur l'image j et non la véritable profondeur du *point 3D* $\mathbf{X}$ lui-même. Cette dernière est notée par $d_j(\mathbf{X})$ (voir plus haut et la figure 1). Comme $\mathbf{X}$ est issu de la carte de profondeur de l'image i, les deux profondeurs seront, bien sûr, identiques pour cette image, $\mathcal{D}_i(\mathbf{X}) = d_i(\mathbf{X})$.

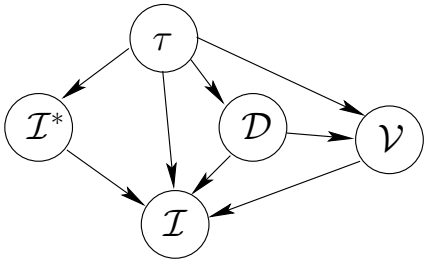

FIG. 2 – *Réseau représentant la décomposition de la distribution de probabilité conjointe. L'absence de flèches entre deux variables représente leur indépendance conditionnelle.*

Pour pouvoir trouver le *meilleur* modèle il faut comparer les images d'entrée avec les images prédites à partir du modèle. La nature de notre modèle, un nuage de points, rend difficile la détermination de la couleur que le modèle prédit pour un certain pixel. Plusieurs points du modèle (au moins un) peuvent se projeter sur le même pixel, mais tous ne sont pas visibles à cause des occultations géométriques. De plus, il se peut que la valeur d'un pixel soit due à une réflexion spéculaire ou à d'autres effets et non pas à la véritable couleur de la scène.
Suivant la solution de Strecha et al. [14], nous introduisons une variable booléenne $\mathcal{V}_{i,\mathbf{X}}$ pour chaque point du modèle $\mathbf{X}$ et chaque image $\mathcal{I}_i$, qui indique si $\mathbf{X}$ est visible ou non dans l'image $\mathcal{I}_i$. Ces variables sont cachées et seulement leurs distributions de probabilité seront calculées.

2.2 Décomposition

Maintenant que toutes les variables sont définies, il nous faut choisir une décomposition de leur distribution de probabilité conjointe. La décomposition définit les dépendances et indépendances conditionnelles entre les variables.
Pour être complet, nous ajoutons aux variables définies précédemment, une variable, $\tau = \{\Sigma, \sigma, \sigma', v, l\}$, qui représente l'ensemble des paramètres qui seront utilisés dans nos formules (voir plus bas). La probabilité conjointe de l'ensemble des variables est donc $p(\mathcal{I}, \mathcal{V}, \mathcal{I}^*, \mathcal{D}, \tau)$ et la décomposition proposée est :

$$p(\tau)\ p(\mathcal{I}^*|\tau)\ p(\mathcal{D}|\tau)\ p(\mathcal{V}|\mathcal{D},\tau)\ p(\mathcal{I}|\mathcal{V},\mathcal{I}^*,\mathcal{D},\tau) \quad (1)$$

La figure 2 représente cette décomposition sous la forme d'un réseau. Chaque terme du produit correspond à une flèche ou une feuille du réseau.

1. $p(\tau)$ est la probabilité *a priori* des paramètres. Ici, nous adoptons un *a priori* uniforme et nous ignorons ce terme.
2. $p(\mathcal{I}^*|\tau)$ est la probabilité *a priori* des couleurs estimées. Ce terme est utilisé avec succès par Fitzgibbon et al. [5] pour régulariser le problème de la génération de nouvelles vues. Un *a priori* à base d'images naturelles est utilisé pour forcer les images estimées $\mathcal{I}^*$ à avoir un aspect d'image naturelle. En pratique, ceci est fait en utilisant un catalogue d'exemples d'imagettes [6]. Dans ce travail, nous utilisons un *a priori* uniforme, concentrant les efforts de régularisation sur les cartes de profondeur.
3. $p(\mathcal{D}|\tau)$ est l'*a priori* sur les cartes de profondeur. Son rôle est de lisser et fusionner les différentes cartes de profondeur. Nous le développerons plus en détail dans la section 2.5. Notez que contrairement à [14], nous ne modélisons aucune dépendance statistique entre $\mathcal{I}^*$ et $\mathcal{D}$. Cela pourrait être utile pour traiter des surfaces avec albédo constant, où les discontinuités des images et celles des cartes de profondeur sont corrélées. En revanche, ceci peut empêcher le lissage des surfaces lisses mais richement texturées.
4. $p(\mathcal{V}|\mathcal{D},\tau)$ est l'*a priori* sur les variables de visibilité. Nous proposons de considérer la visibilité comme étant dépendante des profondeurs $\mathcal{D}$, pour permettre des raisonnements géométriques sur les occultations (section 2.4). Dans l'étape E de l'algorithme EM décrit plus tard, cet *a priori* géométrique est mélangé avec l'information photométrique pour donner une estimation de la visibilité qui est plus robuste aux occultations géométriques.
5. $p(\mathcal{I}|\mathcal{V},\mathcal{I}^*,\mathcal{D},\tau)$ est la vraisemblance du modèle par rapport aux images d'entrée. Une attention particulière est accordée à ce terme (section 2.3), car la formulation usuelle n'est pas satisfaisante dans le cas où les points de vue sont écartés.

Les variables peuvent être classifiées en trois groupes : les variables connues (les données) $\mathcal{I}$ et τ, les variables recherchées (le modèle) $\theta = \{\mathcal{I}^*, \mathcal{D}\}$ et celles cachées $\mathcal{V}$. Notre problème d'inférence consiste alors à trouver la valeur la plus probable des variables recherchées, sachant la valeur des variables connues et marginalisant celles cachées. Nous voulons, donc, estimer

$$\begin{aligned}\theta^* &= \arg\max_\theta p(\theta|\mathcal{I},\tau) \\ &= \arg\max_\theta \int p(\mathcal{I},\mathcal{V},\mathcal{I}^*,\mathcal{D},\tau)d\mathcal{V}\end{aligned}$$

Les prochaines sections donnent une formulation à chacun des termes de la décomposition.

2.3 La vraisemblance

Les pixels des images d'entrée sont considérées comme des observations bruitées du modèle. Nous supposons que le bruit sur les pixels est indépendamment et identiquement distribué. La vraisemblance peut alors être décomposée en un produit des vraisemblances de tous les pixels :

$$p(\mathcal{I}|\mathcal{V},\theta) = \prod_i \prod_{\mathbf{x}} p(\mathcal{I}_i(\mathbf{x})|\mathcal{V},\theta) \quad (2)$$

Ce produit s'étend sur les pixels des images d'entrée et non sur les points du modèle contrairement à l'équation souvent

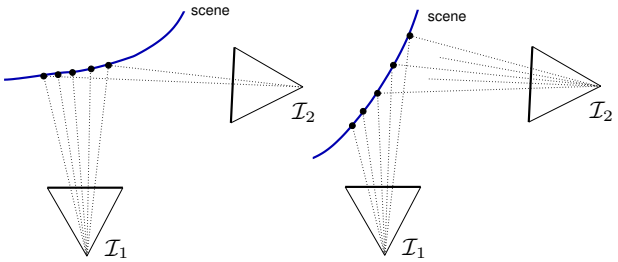

FIG. 3 – *À gauche, plusieurs points 3D provenant de la première carte de profondeur se projettent sur le même pixel de la deuxième image. À droite, pour plusieurs pixels de la deuxième image il n'y a aucun point 3D de la première carte de profondeur qui s'y projette.*

utilisée dans des travaux précédents,

$$p(\mathcal{I}|\mathcal{V},\theta) = \prod_{\mathbf{X}} \prod_{i} p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\mathcal{V}) \qquad (3)$$

Cette expression a le gros avantage de représenter clairement l'apport de chaque point 3D du modèle à la vraisemblance totale, mais elle est malheureusement incorrecte.
La figure 3 esquisse les problèmes qui émergent d'une telle approximation. Dans le premier cas, plusieurs points 3D générés par la carte de profondeur de la première image se projettent sur le même pixel de la deuxième image. Si l'on calcule la vraisemblance en utilisant le produit (3), nous utilisons ce pixel plusieurs fois. Ceci n'est pas souhaitable, car ce pixel voit la surface par un angle rasant et, par conséquent, sa couleur est assez aléatoire et dépend des capteurs de la caméra. Inversement, dans le deuxième cas, c'est la première image qui voit la surface par un angle rasant. Les projetés des points de la première carte sur la deuxième image ne couvrent pas tous ses pixels. Plusieurs pixels de la deuxième image ne seront pas utilisés, ce qui est dommageable étant donné que c'est elle maintenant qui contient le plus d'information sur cette partie de la scène.
Quand les points de vue sont proches les uns des autres, il y a presque une bijection entre les pixels de chaque image et les points 3D provenant des cartes de profondeur des autres images. Dans ce cas, ces effets n'ont pas trop d'importance et peuvent être ignorés. En revanche, quand les points de vue sont écartés il est important de les éviter. Dans la suite, nous présentons notre approximation de la vraisemblance (2).
La vraisemblance $p(\mathcal{I}_i(\mathbf{x})|\mathcal{V},\theta)$ de chaque pixel mesure la similarité entre la couleur $\mathcal{I}_i(\mathbf{x})$ observée dans le pixel $\mathbf{x}$ dans l'image i et la couleur que le modèle prédit pour ce pixel, que nous appelons dans la suite $C_i^*(\mathbf{x})$. La définition de cette couleur est difficile et correspond à un problème de rendu de nuages de points. Elle doit être calculée à partir de la couleur de tous les points du modèle qui se projettent sur ce pixel. Appelons $S_{i,\mathbf{x}}$ l'ensemble de ces points. Une possibilité est de calculer $C_i^*(\mathbf{x})$ comme la couleur moyenne de tous les points visibles ($\mathcal{V}_{i,\mathbf{X}} = 1$) de $S_{i,\mathbf{x}}$. Malheureusement, cette définition n'est pas facile à utiliser : les dérivées de la vraisemblance par rapport aux positions des points sont difficiles à calculer.
Pour approcher cette définition par une expression plus simple nous définissons la vraisemblance d'un pixel comme la moyenne géométrique des vraisemblances que le pixel aurait eues par rapport à chacun des points de $S_{i,\mathbf{x}}$:

$$p(\mathcal{I}_i(\mathbf{x})|\mathcal{V},\theta) = \prod_{\mathbf{X}\in S_{i,\mathbf{x}}} p(\mathcal{I}_i(\mathbf{x})|C(\mathbf{X}),\Sigma)^{\frac{1}{|S_{i,\mathbf{x}}|}}.$$

La moyenne géométrique des probabilités est équivalente à la moyenne arithmétique des énergies. L'idée est de découper l'information du pixel en $|S_{i,\mathbf{x}}|$ morceaux et d'en donner un à chaque point de $S_{i,\mathbf{x}}$. Ceci permet d'utiliser tous les points de $S_{i,\mathbf{x}}$ sans sur-utiliser le pixel $\mathbf{x}$. Cette approximation heuristique permet d'écrire l'équation de la vraisemblance (2) comme un produit sur les points du modèle :

$$p(\mathcal{I}|\mathcal{V},\theta) = \prod_{\mathbf{X}} \prod_{i} p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\Sigma)^{\frac{1}{|S_{i,\mathrm{P}_i\mathbf{X}}|}} \qquad (4)$$

Nous appellerons le terme $p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\Sigma)$ la *vraisemblance point-pixel*. Elle sera modélisée comme un mélange d'une distribution normale et d'une distribution uniforme. Si le point est visible, $\mathcal{V}_{i,\mathbf{X}} = 1$, la vraisemblance de la couleur du pixel sera une distribution normale centrée sur la couleur du point $C(\mathbf{X})$ et avec une variance Σ (la variance du bruit). Si le point n'est pas visible, $\mathcal{V}_{i,\mathbf{X}} = 0$, la vraisemblance sera une distribution uniforme dans l'espace de couleurs.

$$\begin{aligned} p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\Sigma) &= p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{D})\mathcal{N}(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\Sigma) \\ &+ p(\mathcal{V}_{i,\mathbf{X}} = 0|\mathcal{D})\mathcal{U}(\mathcal{I}_i(\mathbf{X})) \end{aligned} \qquad (5)$$

Si l'*a priori* sur les variables de visibilité est constant, cette distribution est appelée une *Gaussienne contaminée* [17]. La prochaine section présente la forme que nous proposons pour cet *a priori*.

2.4 L'a priori géométrique sur la visibilité

Le mélange de la vraisemblance point-pixel (5) est pondéré par l'*a priori* sur la visibilité $p(\mathcal{V}_{i,\mathbf{X}}|\mathcal{D})$. Ce terme représente notre information *a priori* sur la visibilité du point $\mathbf{X}$ dans l'image $\mathcal{I}_i$, avant qu'on ne prenne en compte les couleurs $C(\mathbf{X})$ et $\mathcal{I}_i(\mathbf{x})$. Une distribution uniforme est souvent utilisée [14, 18]. Par contre, notre décomposition (1) de la probabilité conjointe, permet d'utiliser les cartes de profondeur pour savoir s'il est occulté.
$\mathcal{D}_i(\mathbf{X})$ est la profondeur estimée du pixel $\mathbf{x}$ de l'image $\mathcal{I}_i$ dans lequel se projette $\mathbf{X}$. Comme nous l'avons déjà mentionné, elle est en général différente de la profondeur $d_i(\mathbf{X})$ de $\mathbf{X}$ par rapport à cette image. Si $d_i(\mathbf{X})$ est similaire à $\mathcal{D}_i(\mathbf{X})$, le point $\mathbf{X}$ est proche du point correspondant au pixel $\mathbf{x}$. Il semble probable, alors, que $\mathbf{X}$ soit visible dans $\mathbf{x}$. Réciproquement, si $d_i(\mathbf{X})$ est très différente de $\mathcal{D}_i(\mathbf{X})$, on n'admettra pas que $\mathbf{X}$ soit visible : il est loin du point

3D correspondant à $\mathbf{x}$. Grâce à cette observation, la visibilité peut être gérée d'une façon très efficace.
Dans [18] un seuil était utilisé pour déterminer la visibilité d'une façon stricte. Ici nous quantifions les idées précédentes par la formule (souple) :

$$p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{D}) = v \exp\left(-\frac{(d_i(\mathbf{X}) - \mathcal{D}_i(\mathbf{X}))^2}{2\sigma^2}\right)$$

où $v \in [0,1]$ est l'*a priori* pour les points qui sont à la profondeur estimée $\mathcal{D}_i(\mathbf{X})$ et σ est un paramètre qui règle la tolérance aux différences de profondeurs.
Cet *a priori* a l'effet désiré sur la vraisemblance pixel-point. Pour les points proches de la profondeur estimée $\mathcal{D}_i(\mathbf{X})$, l'*a priori* est grand et la distribution normale centrée sur $C(\mathbf{X})$ du mélange (5) est plus fortement pondérée. Ceci rend les couleurs similaires à $C(\mathbf{X})$ plus probables. Pour les points qui sont loins de la profondeur $\mathcal{D}_i(\mathbf{X})$, la distribution uniforme est privilégiée et leur couleur $C(\mathbf{X})$ est ignorée.

2.5 A priori sur les cartes de profondeur

L'*a priori* sur les cartes de profondeur $p(\mathcal{D}|\tau)$ doit évaluer la plausibilité d'un ensemble de cartes sans aucune autre information que les cartes elles-mêmes. Deux propriétés sont désirables :

1. Chacune des cartes de profondeur doit être majoritairement lisse, mais des fortes discontinuités doivent néanmoins être permises.
2. Les points 3D provenant des différentes cartes doivent se mélanger pour générer une seule surface.

Au lieu de quantifier ces deux critères par deux termes séparés, on évaluera les deux par une seule expression. Pour ceci, nous regardons l'ensemble des cartes de profondeur comme un nuage de points oubliant, pour l'instant, les relations de voisinage entre les points 3D issus de pixels voisins d'une même image. Le lissage et la superposition des cartes seront réalisés en forçant les points à s'attirer mutuellement, indépendamment du fait qu'ils soient originaires de la même carte ou pas. Ceci est formalisé comme suit.
Nous exprimons l'*a priori* sur le nuage de points sous la forme d'un réseau de Markov :

$$p(\mathcal{D}) \propto \prod_{\mathbf{X}\in\mathcal{D}} \prod_{\mathbf{Y}\in N(\mathbf{X})} \varphi(\mathbf{X},\mathbf{Y}) \qquad (6)$$

où $N(\mathbf{X})$ est un voisinage de $\mathbf{X}$ et $\varphi(\mathbf{X},\mathbf{Y})$ est une mesure de compatibilité de la paire $(\mathbf{X},\mathbf{Y})$. Pour l'instant, le voisinage s'étend à la totalité des points, $N(\mathbf{X}) = \mathcal{D}\backslash\{\mathbf{X}\}$. De la même façon que pour la vraisemblance point-pixel (5), nous modélisons la mesure de compatibilité comme un mélange entre une loi normale et une d'uniforme, pondéré par un processus de ligne caché $\mathcal{L}$:

$$\begin{aligned}\varphi(\mathbf{X},\mathbf{Y}) &\propto p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 1)\,\mathcal{N}(\mathbf{Y}|\mathbf{X},\sigma') \\ &+ p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 0)\,\mathcal{U}(\mathbf{Y})\end{aligned}$$

où $p(\mathcal{L}_{\mathbf{X},\mathbf{Y}})$ est l'*a priori* sur le processus. Nous adoptons un *a priori* uniforme et $l = p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 1)$ est un paramètre de notre méthode. σ' est la variance de la distribution normale (isotropique) $\mathcal{N}$, et $\mathcal{U}$ est une distribution uniforme sur un volume contenant la scène ($\mathcal{U}(\mathbf{Y}) = \mathcal{U}(\mathbf{X})$).
L'idée est que le processus $\mathcal{L}_{\mathbf{X},\mathbf{Y}}$ indique si les deux points $\mathbf{X}$ et $\mathbf{Y}$ doivent s'attirer ou pas. Si $\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 1$, on regarde $\mathbf{Y}$ comme s'il était une mesure bruitée de $\mathbf{X}$ et on mesure sa probabilité par une normale centrée en $\mathbf{X}$ et de variance σ'. Notez que cette relation est symétrique. Si $\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 0$ nous mesurons la compatibilité entre $\mathbf{X}$ et $\mathbf{Y}$ par une loi uniforme pour exprimer le fait que les positions de ces points sont indépendantes.
Cet *a priori* $p(\mathcal{D})$ est lent à calculer. Si m est le nombre de points, il y a $O(m^2)$ mesures de compatibilité. Par contre, pour tous les points assez loin de $\mathbf{X}$, $\mathcal{N}(\mathbf{Y}|\mathbf{X},\sigma')$ sera très petit et $\varphi(\mathbf{X},\mathbf{Y})$ sera presque constant (égal à $p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 0)\mathcal{U}(\mathbf{Y})$). On peut donc réduire le voisinage aux points proches de $\mathbf{X}$. Nous définissons le voisinage comme étant les points contenus par une sphère centrée en $\mathbf{X}$ et de rayon ρ dépendant de σ'. Trouver ce voisinage peut en général déjà être très coûteux. Heureusement, notre nuage de points provient d'un ensemble de cartes de profondeur dans les quelles les points sont ordonnés (via les pixels sous-jacents). La projection de la sphère dans chaque image est une ellipse et les pixels à l'intérieur de cette ellipse est facile à déterminer. L'ensemble des points 3D qui proviennent de ces pixels est alors un sur-ensemble des voisins de $\mathbf{X}$.
Comme nous désirions, cet *a priori* lisse et intègre toutes les cartes en même temps. Les discontinuités sont tolérées grâce aux variables $\mathcal{L}$ qui empêchent les points de s'attirer lorsqu'ils sont trop distants l'un de l'autre.

Kernel Correlation. Notre *a priori* est très lié avec la *leave-one-out kernel correlation*. Tsin et Kanade ont montré l'utilité de l'*a priori* KC pour lisser des cartes de profondeur tout en conservant les discontinuités et ils l'ont appliqué de façon satisfaisante au problème de la stéréo [19]. Cet *a priori* peut s'écrire aussi sous la forme d'un réseau de Markov avec

$$\varphi_{KC}(\mathbf{X},\mathbf{Y}) \propto exp(\mathcal{N}(\mathbf{X}|\mathbf{Y},\sigma'))$$

La figure 4 contient les logarithmes de notre mesure de compatibilité et de celle de la KC, et montre qu'elles ont une forme similaire. L'avantage de l'utilisation d'un mélange est que l'on reste dans le cadre probabiliste, ce qui permet d'ajouter des nouvelles informations. On pourrait par exemple modéliser la dépendance entre la couleur des points et les variables cachées $\mathcal{L}$, pour faire en sorte que les points de couleurs similaires s'attirent plus.

2.6 Optimisation

Nous maximisons la probabilité *a posteriori* par l'algorithme EM (Expectation Maximization) [3]. Il serait possible et moins coûteux d'optimiser l'*a posteriori* directement par une méthode d'optimisation non-linéaire. Par

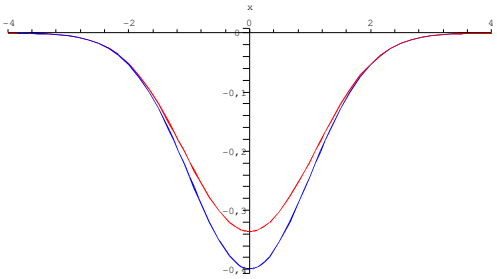

FIG. 4 – *Graphique des potentiels de notre mesure de compatibilité et de celle de la* kernel correlation. *La courbe rouge correspond à notre potentiel* $-\log(\mathcal{N}(x|0,1)-1)$ *et la bleue à celui de la KC* $-\mathcal{N}(x|0,1)$.

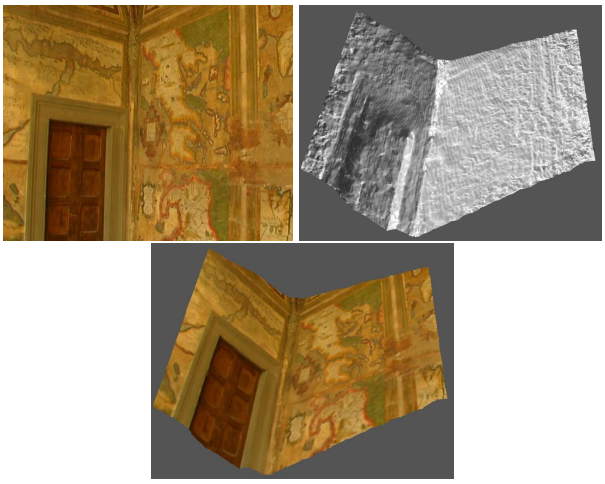

FIG. 5 – **Loggia** : *une des trois images d'entrée et des rendus du modèle obtenu sans et avec texture.*

contre, EM est souvent plus stable et plus facile à suivre car les variables cachées sont explicitement estimées. EM alterne entre l'estimation des probabilités des variables cachées et l'optimisation du modèle. On commence par un modèle initial θ^0 (voir section 3) et on répète les étapes jusqu'à convergence.

E-step. Dans l'étape E, nous calculons les probabilités *a posteriori* des variables cachées $\mathcal{V}$, sachant l'estimation actuelle du modèle. Nous les gardons dans des *cartes de visibilité* $f_{i,\mathbf{X}}$:

$$f_{i,\mathbf{X}} = p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{I}, \theta^t)$$

et, par la règle de Bayes,

$$f_{i,\mathbf{X}} = \frac{p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{D})\mathcal{N}}{p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{D})\mathcal{N} + p(\mathcal{V}_{i,\mathbf{X}} = 0|\mathcal{D})\mathcal{U}}$$

où $\mathcal{N} = \mathcal{N}(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}), \Sigma)$ et $\mathcal{U} = \mathcal{U}(\mathcal{I}_i(\mathbf{X}))$ (voir (5)). C'est ici que l'*a priori* géométrique sur la visibilité est mélangé avec l'évidence photométrique pour donner une estimation de la visibilité.

M-step. Dans l'étape M, nous utilisons les cartes de visibilité pour maximiser le *log-posteriori* espéré,

$$\theta^{t+1} = \arg\max_{\theta} \langle \log p(\mathcal{I}|\mathcal{V}, \theta) \rangle_f + \log p(\mathcal{D})$$

qui est la somme de la log-vraisemblance espérée (voir (4) et (5)),

$$\langle \log p(\mathcal{I}|\mathcal{V}, \theta) \rangle_f = \sum_{\mathbf{X}} \sum_i \frac{1}{S_{i,\mathbf{X}}} (f_{i,\mathbf{X}}\mathcal{N} + (1 - f_{i,\mathbf{X}})\mathcal{U})$$

et le *log-priori* (voir (6)),

$$\log p(\mathcal{D}) = \sum_{\mathbf{X}} \sum_{\mathbf{Y}} \log \varphi(\mathbf{X}, \mathbf{Y})$$

Le maximum est recherché par une méthode de descente de gradient. Les dérivées du *log-posteriori* espéré par rapport aux variables du modèle peuvent être facilement calculées de façon analytique. Dans notre implémentation, une seule itération de la descente du gradient est faite par étape M. Cette itération trouve un meilleur modèle θ^{t+1} mais pas le meilleur (on parle alors d'algorithme EM généralisé). Nous procédons ainsi car chaque itération de la méthode du gradient est aussi coûteuse que l'étape E. Alterner rapidement entre les deux étapes permet d'actualiser les cartes de visibilité plus fréquemment.

3 Expérimentations

Nous avons implémenté l'algorithme dans un schéma pyramidal pour accélérer la vitesse de convergence et agrandir le bassin de convergence. On commence par utiliser des versions réduites des images d'entrée, et des cartes. Une fois que EM converge, un niveau à une resolution plus grande est initialisé avec les résultats obtenus en utilisant une interpolation bilinéaire.

Nous décrivons d'abord comment nous avons choisi les paramètres $\tau = \{\Sigma, \sigma, \sigma', v, l\}$ de notre méthode. Dans toutes nos expériences, la variance du bruit des images Σ (voir section 2.3) était incluse dans l'ensemble des variables recherchées, et elle était donc estimée pendant l'optimisation. Si un point est à la profondeur estimée, il est très probable qu'il soit visible, c'est pourquoi nous avons choisi la valeur 0.9 pour le paramètre v (voir section 2.4). La valeur de σ' était toujours égale à celle de σ (voir sections 2.4 et 2.5). Cette valeur était calculée de façon heuristique en faisant la moyenne des distances entre les paires de points 3D voisins dans les images. Le paramètre l (voir section 2.5) est le seul que nous avons modifié spécialement pour chaque expérience. Nous présentons dans la suite, les résultats obtenus sur des jeux de données de difficulté croissante.

Facile. Les données de la Loggia (figure 5) consistent en 3 images prises de points de vue distants d'une scène richement texturée, et avec une géométrie simple. Les cartes de profondeur était initialisées avec une valeur constante, ce

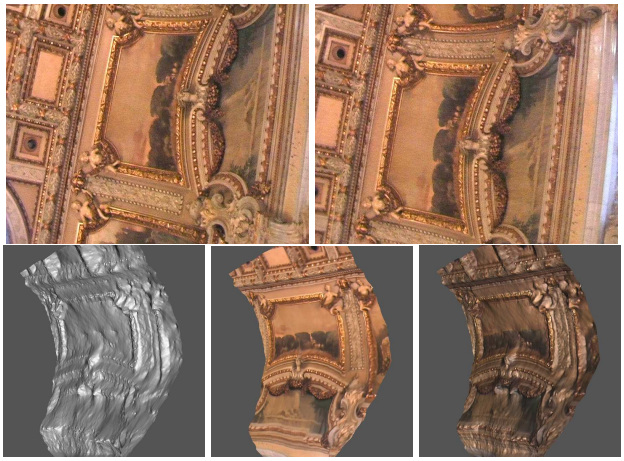

FIG. 6 – **Casino** : *deux des cinq images d'entrée en haut. Des rendus sans texture, avec texture et ré-éclairé de la surface reconstruite en bas.*

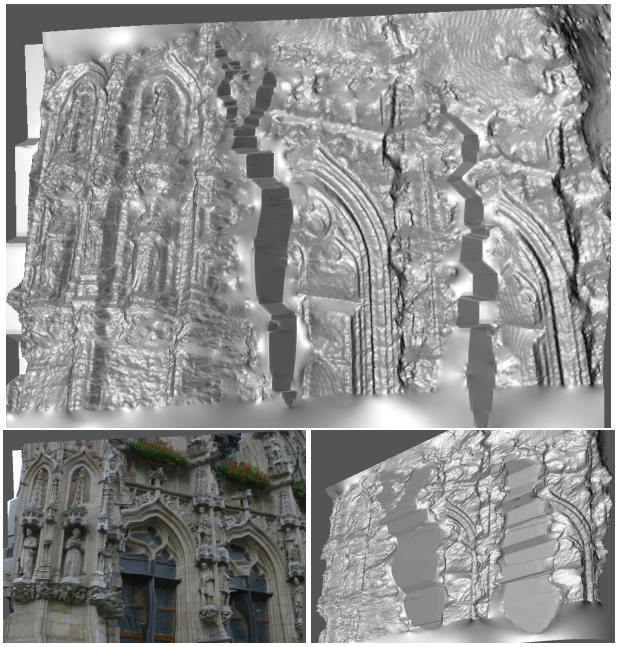

FIG. 7 – **Cityhall** : *rendus sans texture, avec texture et ré-éclairé d'une des cartes de profondeur vue de deux points de vue différents. Tous les points du modèle sont montrés. La partie lisse du bas du modèle correspond aux points qui sont vus par une seule image (la reconstruction de cette partie de la surface ne repose donc que sur l'*a priori *(2.5)). Les parties planes du centre correspondent aux discontinuités de la carte de profondeur (occultations dans les images d'entrée).*

qui correspond à un plan fronto-parallèle par image. L'algorithme a retrouvé la surface correcte.

Les données du Casino (figure 6) contiennent cinq images avec des points de vue proches. L'initialisation était aussi une constante. Les résultats montrent le potentiel de la technique pour retrouver des détails fins. Dans les deux cas, des valeurs suffisamment grandes de l ($l > 0.1$) donnaient des résultats similaires.

Moyen. Nous avons testé la performance de la méthode sur les images de la Cityhall pour prouver qu'elle est au niveau de l'état de l'art. Nous avons utilisé les images 3,4 et 5 [1]. Cette fois, nous avons initialisé le modèle à partir des positions 3D des points caractéristiques calculés dans l'étape de calibrage. La profondeur de ces points reste fixée pendant que l'on lisse le reste des pixels des cartes de profondeur avec un filtre Gaussien. Le résultat est un ensemble de surfaces (une par carte) lisses qui passent par les points caractéristiques. Avec cette initialisation grossière, l'algorithme a convergé, fusionnant les différentes cartes dans une seule surface. Les résultats (figure 7) montrent des détails très fins. La grosse discontinuité entre les statues du premier plan et la porte du fond est préservée.

Difficile. Pour montrer la capacité de notre *a posteriori* de préserver les discontinuités et gérer les occultations géométriques, nous l'avons testé avec les difficiles images de la statue (voir figures 8 et 9). La scène contient une statue du premier plan et un mur au fond. Une seule carte de profondeur n'est pas assez pour modéliser cette scène car aucune des images contienne la totalité de la statue ou du mur. Nous avons initialisé le modèle par la même procédure que pour la Cityhall.

La difficulté principale a été d'estimer précisément la très grosse discontinuité entre la statue et le mur. Lisser dans cette région aurait produit des points 3D incorrects, situés entre le premier plan et le fond. Nous avons réglé le paramètre l à une valeur petite ($l = 0.2$) pour motiver les points à ne pas trop s'attirer (voir section 2.5). La discontinuité est bien préservée, mais pas exactement au bon endroit. Quelques points du fond sont restés attachés à la statue. De plus, pour initialiser un niveau plus détaillé de la pyramide à partir d'un de plus grossier, nous avons utilisé une interpolation bilinéaire, ce qui a encore lissé la discontinuité.

Pour résoudre ces problèmes, nous avons alterné quelques itérations de EM avec l'heuristique de recherche globale suivant : pour chaque pixel $\mathbf{x}$ d'une image donnée i, on considère toutes les profondeurs de tous les points 3D qui se projettent sur ce pixel (voir section 2.3). On teste alors si la vraisemblance serait meilleure si l'on échangeait la profondeur de ce pixel par une de ces valeurs et on garde la profondeur qui produit la meilleure vraisemblance. La grosse discontinuité entre la statue et le mur était détectée, par EM dès le niveau le plus grossier. L'heuristique a placé cette discontinuité au bon endroit et l'a conservée dans les niveaux suivants.

[1] Les images de la Cityhall et leur calibrage peuvent être téléchargés de `http://www.esat.kuleuven.ac.be/~cstrecha/testimages/`

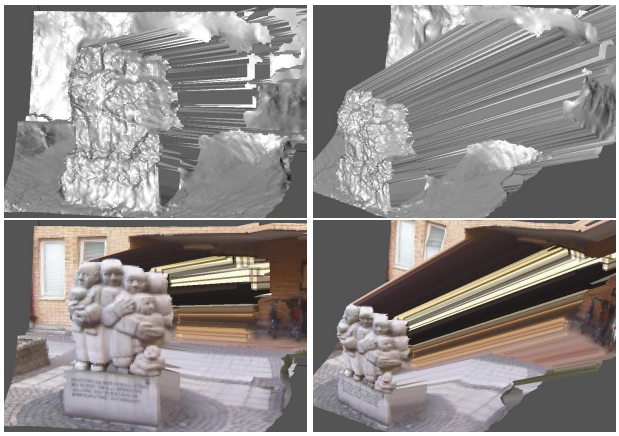

FIG. 8 – **Statue** : *quelques rendus de la carte de profondeur $\mathcal{D}_2$. Les lignes correspondent aux fortes discontinuités entre la statue et le fond.*

4 Discussion

Nous avons proposé une méthode pour construire des modèles 3D d'une scène à partir d'images en utilisant plusieurs cartes de profondeur colorées. Chacun des pixels des images d'entrée est représenté dans le modèle. Par conséquent, il est possible d'obtenir des modèles de haute résolution. Néanmoins, il reste encore quelques problèmes à résoudre à fin de rendre la méthode utilisable.

Le cadre probabiliste permet d'apprendre les paramètres au cours de l'optimisation. Effectivement, si l'on traite les paramètres comme des variables aléatoires on peut, soit estimer leurs valeurs les plus probables, soit les marginaliser. Dans notre implémentation, seulement Σ (voir section 2.3) est estimé, les quatre autres paramètres sont réglés à la main. Même si ces paramètres sont faciles à régler parce qu'ils représentent des concepts bien définis, il serait préférable que l'algorithme les trouve lui-même.

Un autre problème qui reste à résoudre est l'initialisation. L'implémentation pyramidale de EM converge correctement dans les cas où les fortes discontinuités sont capturées dès les premiers niveaux. Cependant, sans une bonne initialisation, il est assez clair que pour des images comme celles qui sont utilisées dans [13], l'algorithme ne trouvera pas l'optimum global mais un de local. Curieusement, une des méthodes qui marche le mieux pour ces images [16], utilise le même formalisme Bayésien, mais un autre algorithme d'optimisation. Au lieu de chercher le modèle le plus probable, elle utilise l'algorithme *Loopy Belief Propagation* pour calculer la distribution de probabilité *a posteriori*. Dans un avenir proche nous voulons appliquer un algorithme similaire sur notre définition de l'*a posteriori*.

Références

[1] P.N. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3) :237–260, 1996.

[2] A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *ICCV*, volume 1, page 388, 2001.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B*, 39 :1–38, 1977.

[4] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *ECCV*, pages 379–393, 1998.

[5] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *ICCV*, pages 1176–1183, 2003.

[6] W. T. Freeman and E. C. Pasztor. Learning low-level vision. *IJCV*, 40 :25 – 47, 2000.

[7] P. Fua and Y. Leclerc. Object-centered surface reconstruction : combining multi-image stereo shading. In *Image Understanding Workshop*, pages 1097–1120, 1993.

[8] V. Kolmogorov, R. Zabih, and S. J. Gortler. Generalized multi-camera scene reconstruction using graph cuts. In *EMMCVPR*, pages 501–516, 2003.

[9] K. Kutulakos and S. Seitz. A theory of shape by space carving. *IJCV*, 38(3) :199–218, 2000.

[10] D. Morris and T. Kanade. Image-consistent surface triangulation. In *CVPR*, pages 332–338, 2000.

[11] S. Paris, F. Sillion, and L. Quan. A surface reconstruction method using global graph cut optimization. In *Asian Conference of Computer Vision*, January 2004.

[12] M. Pollefeys. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. PhD thesis, Katholieke Universiteit Leuven, Belgium, May 1999.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47 :7–42, 2002.

[14] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views : a probabilistic account. In *CVPR*, volume 1, pages 552–559, 2004.

[15] C. Strecha, T. Tuytelaars, and L. Van Gool. Dense matching of multiple wide-baseline views. In *ICCV*, volume 2, pages 1194–1201, 2003.

[16] J. Sun, H.Y. Shum, and N.N. Zheng. Stereo matching using belief propagation. *PAMI*, 25(7), July 2003.

[17] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3) :271–301, 1990.

[18] R. Szeliski. A multi-view approach to motion and stereo. In *CVPR*, pages 1157–1163, 1999.

[19] Y. Tsin and T. Kanade. A correlation-based model prior for stereo. In *CVPR*, volume 1, pages 135–142, 2004.

[20] G. Vogiatzis, P.H.S. Torr, and R. Cipolla. Bayesian stochastic mesh optimization for 3d reconstruction. In *Proceedings 14th British Machine Vision Conference*, 2003.

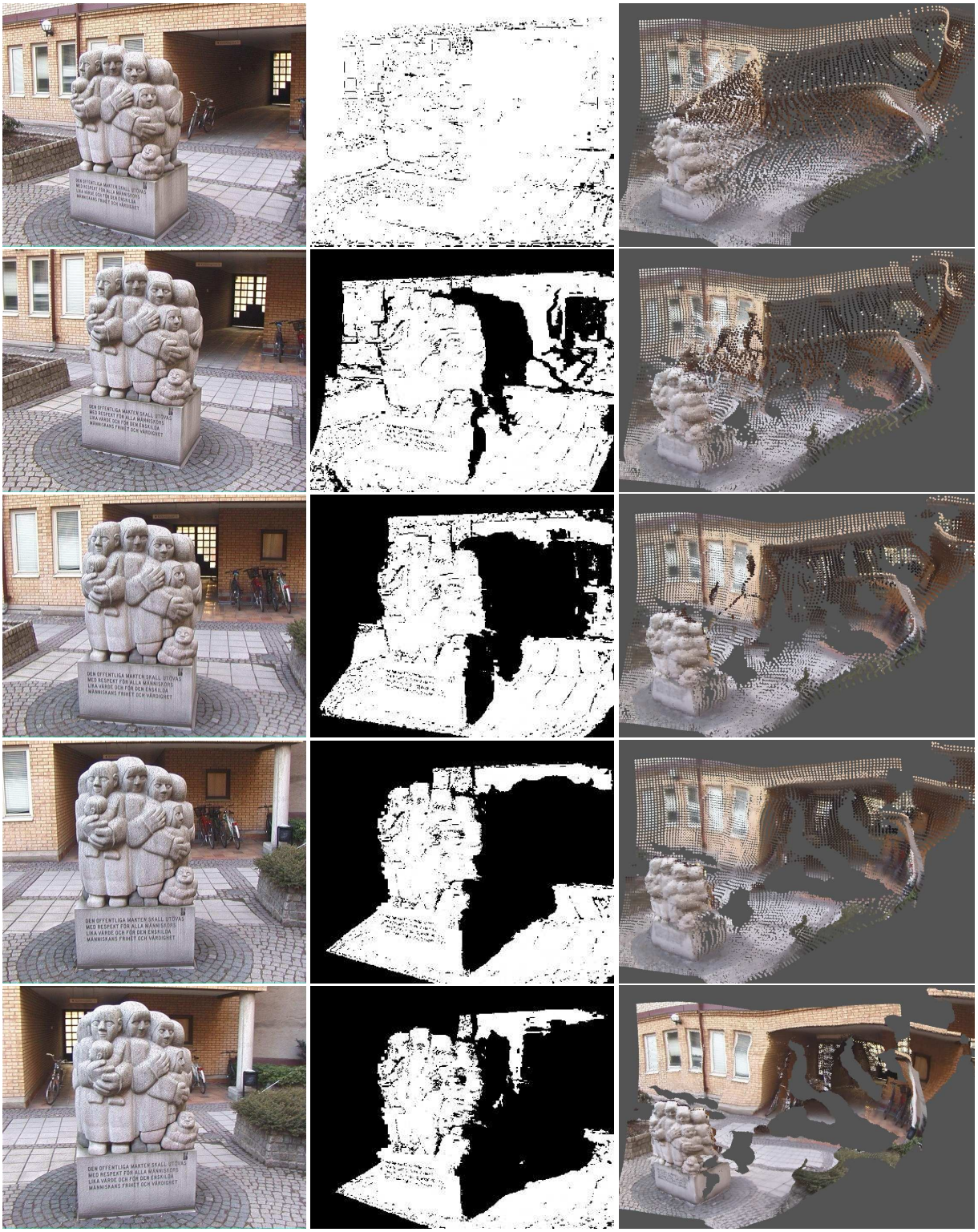

FIG. 9 – **La statue** : *la première ligne présente les cinq images d'entrée. La seconde, les cartes de visibilité associées à la première image : pour chaque point 3D provenant de la carte de profondeur de la première image il est montré s'il a été estimé d'être visibble sur les 5 images d'entrée au pas. Un pixel qui est blanc dans l'image de la i^e colonne de cette figure veut dire que le point 3D associé, $\mathbf{X}_1(\mathbf{x}, \mathcal{D}_1(\mathbf{x}))$, a été estimé comme étant visible par rapport à l'image $\mathcal{I}_i$. La dernière ligne montre des rendus à base de points de l'ensemble de cartes de profondeur pendant l'évolution de l'algorithme, de l'initialisation jusqu'au modèle final.*